

La metrica Autoregressiva tra modelli ARIMA: una procedura in linguaggio GAUSS

Marcella Corduas

Dipartimento di Scienze Statistiche, Universita' di Napoli Federico II
Centro di Specializzazione e Ricerche, Portici (NA)
E-mail: corduas@unina.it

Summary: In this article we discuss the statistical properties of the Autoregressive distance between ARIMA models. This criterion has been proposed to measure structural diversity between time series. It has been applied in several fields such as time series classification and seasonal adjustment.

The attention is focussed on the computational aspects related to the use of the Autoregressive distance. We present two procedures which allow, respectively, the computation of the AR distance, the representation of the time series models through the Multidimensional Scaling, the performance of a test of hypothesis to check if two time series have been generated by the same linear stochastic process.

Keywords: Autoregressive metric, ARIMA models, Time series classification

1. Introduzione

La ricerca di strumenti statistici per il confronto di dati dinamici e' un tema che ha riscosso l'attenzione degli studiosi di serie storiche in vari settori scientifici e, a tal fine, ha generato criteri di distanza alternativi.

In questo lavoro viene presa in considerazione la metrica Autoregressiva, proposta da Piccolo (1984a), mediante la quale e' possibile confrontare le serie storiche attraverso le formulazioni parametriche dei corrispondenti modelli ARIMA.

In particolare, presenteremo una procedura implementata in linguaggio GAUSS, che consente l'analisi di un insieme di serie storiche a fini di classificazione. Nel seguito, descriveremo brevemente gli elementi essenziali connessi alla istituzione della metrica Autoregressiva sulla classe dei processi ARIMA e le proprietà asintotiche della distribuzione di una opportuna trasformazione di tale metrica, discutendo infine gli aspetti computazionali connessi alla relativa utilizzazione.

2. La metrica Autoregressiva tra modelli ARIMA

Consideriamo la classe \mathcal{L} dei processi stocastici invertibili $Z_t \sim ARIMA(p, d, q)(P, D, Q)_s$, tali che:

$$Z_t \in \mathcal{L} \Leftrightarrow \phi(B)\Phi(B^s) \nabla^d \nabla_s^p Z_t = \theta(B)\Theta(B^s) a_t \quad (1)$$

dove $a_t \sim WN(0, \sigma^2)$ e' un processo stocastico definito *White Noise* (WN), cioè una successione di variabili casuali di media zero, incorrelate e omoschedastiche. Inoltre, i polinomi nell'operatore B , tale che $B^k Z_t = Z_{t-k}$, $k = 0, \pm 1, \pm 2, \dots$, sono definiti da:

$$\varphi(B) = \phi(B)\Phi(B^s) = (1 - \phi_1 B - \dots - \phi_p B^p)(1 - \Phi_1 B^s - \dots - \Phi_P B^{Ps})$$

$$\vartheta(B) = \theta(B)\Theta(B^s) = (1 - \theta_1 B - \dots - \theta_q B^q)(1 - \Theta_1 B^s - \dots - \Theta_Q B^{Qs}).$$

Tali operatori possiedono tutte le radici esterne al cerchio unitario e non hanno fattori comuni (Box e Jenkins, 1976).

Per questo, qualsiasi processo $Z_t \in \mathcal{L}$ ammette la rappresentazione AR(∞):

$$\pi(B)Z_t = a_t \quad (2)$$

essendo:

$$\pi(B) = (1 - B)^d (1 - B^s)^D \varphi(B) \vartheta^{-1}(B) = 1 - \sum_{i=1}^{\infty} \pi_i B^i.$$

Ora, i coefficienti Autoregressivi convergono in modulo e quindi al quadrato, per cui $\sum_{i=1}^{\infty} |\pi_i| < \infty$ e $\sum_{i=1}^{\infty} \pi_i^2 < \infty$. Inoltre, se a_t e' un processo gaussiano, fissati i valori iniziali e gli ordini del modello ARIMA, la conoscenza di $\{\varphi(B), \vartheta(B), \sigma^2\}$, e poi di $\{\pi(B), \sigma^2\}$, caratterizza completamente la struttura probabilistica del processo stocastico Z_t .

Partendo da tale considerazione, Piccolo (1984a; 1990) ha introdotto una misura di dissimilarita' strutturale tra i processi appartenenti ad \mathcal{L} definendo la seguente metrica:

$$d(X_t, Y_t) = \left\{ \sum_{i=1}^{\infty} (\pi_{x,i} - \pi_{y,i})^2 \right\}^{1/2}. \quad (3)$$

Essa ha un'interpretazione in termini di funzione di previsione, poiche', a parita' di valori iniziali, X_t e Y_t ammetteranno distanza nulla se le previsioni per gli istanti futuri, basate sulla conoscenza fino al tempo t , coincidono.

Nelle applicazioni reali, per serie generate da processi gaussiani (o per serie che -dopo convenienti trasformazioni- possono essere ritenute essere state generate da processi gaussiani), lo stimatore \hat{d} e' ottenuto sostituendo nella (3) gli stimatori di massima verosimiglianza (ML) per i parametri $\{\pi_i\}$ associati a ciascun modello. Nel seguito, faremo riferimento allo stimatore \hat{d}^2 e ne deriveremo la distribuzione asintotica nel caso in cui i processi X_t e Y_t siano indipendenti.

3. Formulazione matriciale per i coefficienti Autoregressivi

Lo studio della distribuzione asintotica di \hat{d}^2 e' stato affrontato in precedenti lavori nel caso di processi stazionari (Corduas, 1992b; 1993; 1996; Maharaj, 1996; Piccolo, 1989) e non-stazionari (Corduas e Piccolo, 1996). La bonta' dei risultati asintotici e' stata verificata

mediante studi di simulazione per le strutture piu' elementari, derivando una adeguata approssimazione di piu' immediato utilizzo.

Qui riprenderemo le conclusioni di tali contributi estendendole alla forma piu' generale di modelli ARIMA con operatori stagionali moltiplicativi. Proporremo, in particolare, una formulazione che consente di sviluppare agevolmente i relativi calcoli.

Per un processo $Z_t \sim ARIMA(p, d, q)(P, D, Q)_s$ indichiamo con $\beta' = \{\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q, \Phi_1, \dots, \Phi_P, \Theta_1, \dots, \Theta_Q\}$ il vettore dei parametri che -assieme alla varianza σ^2 del processo WN- caratterizza il processo gaussiano. I coefficienti Autoregressivi della parte ARMA del modello: $\pi^A = \{\pi_i^A\}$, sono univocamente determinati dalla relazione: $\phi(B)\Phi(B^s) = \theta(B)\Theta(B^s)\pi^A(B)$. Ad essi, sono legati i coefficienti $\pi = \{\pi_i\}$ del polinomio AR(∞) che caratterizza il modello ARIMA, essendo:

$$\pi(B) = \pi^A(B)(1 - B)^d(1 - B^s)^D.$$

Ora, per il calcolo dei coefficienti Autoregressivi si puo' ricorrere ad una formulazione matriciale. A tal fine, supponiamo di troncare opportunamente la successione di coefficienti Autoregressivi ad un certo lag m : tale convenzione, oltre che necessaria per gli aspetti numerici, e' lecita dal punto di vista formale poiche' la convergenza dei coefficienti π_i implica che essi da un certo punto in poi saranno trascurabili. Se, allora, limitiamo l'attenzione alla parte ARMA del modello e poniamo $\pi^A = (\pi_0^A, \pi_1^A, \dots, \pi_{m-1}^A)'$, con $\pi_0^A = -1$, si puo' facilmente verificare che:

$$\pi^A = \mathbf{L}_{Ns}^{-1} \mathbf{L}_s^{-1} \mathbf{F} \mathbf{v} \quad (4)$$

dove:

i) $\mathbf{v} = (-1, \phi_1, \dots, \phi_p, 0, \dots, 0)'$ e' un vettore colonna di m elementi;

ii) \mathbf{L}_{Ns} e' una matrice triangolare a banda diagonale di dimensioni $(m \times m)$ con elementi tutti nulli ad eccezione della diagonale

principale, che possiede elementi unitari. delle q sottodiagonali, che sono definite da: $l_{i+h,i} = -\theta_h$, per $h = 1, 2, \dots, q$ e $i = 1, 2, \dots, (m - h)$, ovvero:

$$\mathbf{L}_{NS} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ -\theta_1 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -\theta_{q-1} & \dots & \dots & \dots & \dots & 0 \\ -\theta_q & -\theta_{q-1} & \dots & 1 & \dots & 0 \\ 0 & -\theta_q & -\theta_{q-1} & \dots & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

iii) \mathbf{L}_S e' una matrice triangolare a banda diagonale di dimensioni $(m \times m)$ con elementi tutti nulli tranne la diagonale principale, che possiede elementi unitari, e Q sottodiagonali che sono definite da: $l_{i+12h,i} = -\Theta_h$, per $h = 1, 2, \dots, Q$; $i = 1, 2, \dots, (m - 12h)$:

$$\mathbf{L}_S = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ -\Theta_1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & -\Theta_1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & -\Theta_1 & \dots & 0 & 0 & 1 & \dots & 0 \\ \dots & 0 \\ -\Theta_2 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & -\Theta_2 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & -\Theta_2 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots \end{bmatrix}$$

iv) \mathbf{F} e' una matrice triangolare a banda diagonale di dimensioni $(m \times m)$ con elementi tutti nulli tranne la diagonale principale, che

possiede elementi unitari, e P sottodiagonali definite da:
 $f_{i+12h,i} = -\Phi_h$, per $h = 1, 2, \dots, P$; $i = 1, 2, \dots, (m - 12h)$, ovvero:

$$F = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ -\Phi_1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & -\Phi_1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & -\Phi_1 & \dots & 0 & 0 & 1 & \dots & 0 \\ \dots & 0 \\ -\Phi_2 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & -\Phi_2 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & -\Phi_2 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots \end{bmatrix}.$$

La derivazione matriciale dei coefficienti π_j puo' sembrare computazionalmente piu' complessa e ingombrante rispetto agli efficienti algoritmi ricorsivi, generalmente adottati in tale contesto. Ora, anche a prescindere dal fatto che in questo lavoro si utilizza un linguaggio di programmazione che ottimizza l'algebra matriciale nei suoi aspetti computazionali, va sottolineato come l'approccio prescelto consenta di fattorizzare in modo netto i parametri delle due componenti di un modello ARIMA (stagionale e non stagionale, rispettivamente); il che, come vedremo, si rivela particolarmente utile nelle successive elaborazioni.

Se sono noti i coefficienti Autoregressivi π^A della struttura ARMA, e' immediato derivare quelli della corrispondente formulazione ARIMA mediante la seguente trasformazione:

$$\pi = A\pi^A \quad (5)$$

dove A e' una matrice triangolare inferiore di dimensioni $(m \times m)$, la cui generica colonna j -esima presenta elementi cosi' caratterizzati:

$$a_{ij} = \begin{cases} 0, & \text{per } i < j \text{ e } i > j + d + sD; \\ \{\text{coefficienti dello sviluppo di } (1-B)^d(1-B^s)^D\}, & \text{per } j \leq i \leq j + d + sD. \end{cases}$$

Di seguito, riportiamo i casi piu' rilevanti per le applicazioni reali:

i) se $d = 1$, $D = 0$, e quindi: $\nabla Z_t \sim \text{ARMA}$, la matrice \mathbf{A} contiene elementi:

$$a_{ij} = \begin{cases} 1, & i = j; \\ -1, & i = j + 1, j = 1, \dots, m - 1; \\ 0, & \text{altrove}; \end{cases}$$

ovvero:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ 0 & 0 & -1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & & -1 & 1 \end{bmatrix}.$$

ii) se $d = 0$, $D = 1$ ($s = 12$), e quindi: $\nabla_{12} Z_t \sim \text{ARMA}$, la matrice \mathbf{A} contiene elementi:

$$a_{ij} = \begin{cases} 1, & i = j; \\ -1, & i = j + 12, j = 1, \dots, m - 12; \\ 0, & \text{altrove}; \end{cases}$$

ovvero:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -1 & \dots & \dots & \dots & \dots & 0 \\ 0 & -1 & \dots & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

iii) se $d = 1, D = 1 (s = 12)$, e quindi: $\nabla \nabla_{12} Z_t \sim \text{ARMA}$, la matrice \mathbf{A} contiene elementi:

$$a_{ij} = \begin{cases} 1, & i = j; \\ -1, & i = j + 1, j = 1, \dots, m - 1; \\ -1, & i = j + 12, j = 1, \dots, m - 12; \\ 1, & i = j + 13, j = 1, \dots, m - 13; \\ 0, & \text{altrove.} \end{cases}$$

ovvero:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ -1 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -1 & 0 & \dots & \dots & \dots & 0 \\ 1 & -1 & \dots & 1 & \dots & 0 \\ 0 & 1 & -1 & \dots & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

Nel prossimo paragrafo, vedremo come tale formulazione matriciale possa essere efficientemente adoperata per derivare la distribuzione asintotica di \hat{d}^2 .

4. La distribuzione asintotica dello stimatore della distanza

Nel caso di processi Autoregressivi, Piccolo (1989) ha dimostrato che lo stimatore di massima verosimiglianza \hat{d}^2 si distribuisce asintoticamente come una combinazione lineare di variabili casuali Chi-quadrato indipendenti. Partendo da tale conclusione, nel già citato contributo di Corduas (1996), il risultato è stato esteso alla classe dei processi ARMA non stagionali. Di seguito, richiamiamo brevemente tale derivazione generalizzandola al caso di processi integrati e con operatori stagionali moltiplicativi.

Si osservi che i coefficienti Autoregressivi possono essere espressi come: $\pi_i^A = f_i(\boldsymbol{\beta})$, $i = 1, 2, \dots$. Sotto opportune assunzioni di regolarità, si dimostra che il vettore degli stimatori di massima verosimiglianza $\hat{\boldsymbol{\beta}}$ è distribuito asintoticamente come una variabile casuale multinormale, che sigleremo con $\hat{\boldsymbol{\beta}} \stackrel{a}{\sim} \mathcal{N}(\boldsymbol{\beta}, \mathbf{V})$ (si vedano: Box e Jenkins, 1976, 282-284; 325; Brockwell e Davis, 1987, 251-253).

Al riguardo, ricordiamo che -quando le serie a confronto sono di pari¹ numerosità n - la matrice delle varianze e covarianze è:

$$\mathbf{V} = n^{-1} \sigma^2 [\mathbb{E}(\mathbf{U}'\mathbf{U})]^{-1},$$

dove \mathbf{U} è una matrice di dimensioni $(n \times p + q + P + Q)$ le cui

¹ Ovviamente, la metrica Autoregressiva può essere utilizzata anche per il confronto tra serie di differente numerosità purché, negli sviluppi successivi, si tenga conto degli effetti di tali quantità nelle varianze degli stimatori.

colonne contengono le derivate:

$$-\frac{\partial a_t}{\partial \beta_j}, j = 1, \dots, p + q + P + Q.$$

Nel caso generale in cui: $a_t = \theta^{-1}(B)\Theta^{-1}(B^s)\phi(B)\Phi(B^s)W_t$ con $W_t = \nabla^d \nabla_s^D Z_t$, le derivate richieste sono le seguenti:

$$\begin{aligned} \frac{\partial a_t}{\partial \phi_j} &= -\phi^{-1}(B)B^j a_t; & \frac{\partial a_t}{\partial \Phi_j} &= -\Phi^{-1}(B^s)B^{sj} a_t; \\ \frac{\partial a_t}{\partial \theta_j} &= \theta^{-1}(B)B^j a_t; & \frac{\partial a_t}{\partial \Theta_j} &= \Theta^{-1}(B^s)B^{sj} a_t. \end{aligned}$$

Inoltre, dalla (4) avremo che, asintoticamente, $\hat{\pi} \stackrel{a}{\sim} \mathcal{N}(\pi, \mathbf{A}\Omega\mathbf{A}')$ essendo: $\Omega = \mathbf{B}\mathbf{V}\mathbf{B}'$, e dove gli elementi della matrice \mathbf{B} sono definiti da:

$$b_{ij} = \left\{ \frac{\partial f_i(\hat{\beta})}{\partial \hat{\beta}_j} \right\}_{\hat{\beta}=\beta}.$$

Nel seguito, per brevit , porremo $\Sigma = \mathbf{A}\Omega\mathbf{A}'$.

D'altra parte, con riferimento alla (3), si avr  anche che $\hat{\pi}_x \stackrel{a}{\sim} \mathcal{N}(\pi_x, \Sigma_x)$ e $\hat{\pi}_y \stackrel{a}{\sim} \mathcal{N}(\pi_y, \Sigma_y)$, cosicche' la distribuzione asintotica di \hat{d}^2 sotto l'ipotesi nulla $H_0: \pi_x = \pi_y$, (ovvero, $H_0: \beta_x = \beta_y = \beta_0$), potr  essere derivata utilizzando una tecnica standard di trasformazione.

Infatti, essendo i processi X_t e Y_t indipendenti, e' immediato verificare che: $(\hat{\pi}_x - \hat{\pi}_y) \stackrel{a}{\sim} \mathcal{N}(\pi_x - \pi_y, \Sigma_x + \Sigma_y)$, e quindi, sotto H_0 , $(\hat{\pi}_x - \hat{\pi}_y) \stackrel{a}{\sim} \mathcal{N}(\mathbf{0}, 2\Sigma)$. Di conseguenza, considerando la trasformazione:

$$\boldsymbol{\delta} = (2\boldsymbol{\Sigma})^{-1/2}(\hat{\boldsymbol{\pi}}_x - \hat{\boldsymbol{\pi}}_y) \stackrel{a}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

si avra' che:

$$\hat{d}^2 = (\hat{\boldsymbol{\pi}}_x - \hat{\boldsymbol{\pi}}_y)'(\hat{\boldsymbol{\pi}}_x - \hat{\boldsymbol{\pi}}_y) = 2\boldsymbol{\delta}'\boldsymbol{\Sigma}\boldsymbol{\delta} = 2\sum_{j=1}^h \lambda_j \chi_{g_j}^2. \quad (6)$$

Nella (6), λ_j ($j = 1, 2, \dots, h$; $h \leq p + q + P + Q$) sono gli autovalori positivi della matrice $\boldsymbol{\Sigma} = \mathbf{A}\boldsymbol{\Omega}\mathbf{A}'$ e $\chi_{g_j}^2$ sono variabili casuali Chi-quadrato indipendenti con g_j gradi di liberta'. Poiche' nelle situazioni reali e' zero la probabilita' che radici stimate dai dati siano esattamente coincidenti, e' lecito supporre, come noi faremo nel seguito, che $g_j \equiv 1, \forall j = 1, 2, \dots, h$.

Naturalmente, l'approssimazione proposta si fonda sulla distribuzione asintotica degli stimatori $\hat{\boldsymbol{\beta}}$, per cui, essa risultera' meno accurata per quel sottoinsieme dello spazio parametrico di $\boldsymbol{\beta}$ ove le premesse di tale approssimazione vengono meno (Corduas, 1996). Di solito, cio' avviene nelle vicinanze della regione di non-invertibilita' per le componenti MA; ovviamente, cio' e' piu' frequente e crea maggiori problemi quando la numerosita' delle osservazioni e' limitata.

Le precedenti formulazioni risolvono il problema sul piano generale ma la determinazione analitica della matrice \mathbf{B} , tranne poche eccezioni, puo' essere piuttosto laboriosa. Per tale motivo, nel seguito ci porremo il problema della sua computabilita'.

Si osservi che la matrice \mathbf{B} puo' essere espressa in termini dei parametri della parte ARMA del modello e quindi calcolata numericamente con un grado di approssimazione prefissato. Infatti, le colonne della matrice \mathbf{B} sono ottenute derivando la (4) rispetto a $\hat{\boldsymbol{\beta}}$, e ponendo poi $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0$.

In particolare, nell'ordine, tali derivate risultano essere:

$$\begin{aligned}
\text{i)} \quad \left\{ \frac{\partial \hat{\pi}^A}{\partial \hat{\phi}_j} \right\}_{\hat{\beta}=\beta_0} &= \mathbf{L}_{NS}^{-1} \mathbf{L}_S^{-1} \mathbf{F} \Delta_{rj}, \\
&\text{dove: } \Delta_{rj} = \frac{\partial \mathbf{v}}{\partial \phi_j}, \quad j = 1, \dots, p; \\
\text{ii)} \quad \left\{ \frac{\partial \hat{\pi}^A}{\partial \hat{\theta}_j} \right\}_{\hat{\beta}=\beta_0} &= - \mathbf{L}_{NS}^{-1} \Delta_{NSj} \mathbf{L}_{NS}^{-1} \mathbf{L}_S^{-1} \mathbf{F} \mathbf{v}, \\
&\text{dove: } \Delta_{NSj} = \frac{\partial \mathbf{L}_N}{\partial \theta_j}, \quad j = 1, \dots, q; \\
\text{iii)} \quad \left\{ \frac{\partial \hat{\pi}^A}{\partial \hat{\phi}_j} \right\}_{\hat{\beta}=\beta_0} &= \mathbf{L}_{NS}^{-1} \mathbf{L}_S^{-1} \Delta_{Fj} \mathbf{v}, \\
&\text{dove: } \Delta_{Fj} = \frac{\partial \mathbf{F}}{\partial \theta_j}, \quad j = 1, 2, \dots, P; \\
\text{iv)} \quad \left\{ \frac{\partial \hat{\pi}^A}{\partial \hat{\Theta}_j} \right\}_{\hat{\beta}=\beta_0} &= - \mathbf{L}_{NS}^{-1} \mathbf{L}_S^{-1} \Delta_{Sj} \mathbf{L}_S^{-1} \mathbf{F} \mathbf{v}, \\
&\text{dove: } \Delta_{Sj} = \frac{\partial \mathbf{L}_S}{\partial \Theta_j}, \quad j = 1, 2, \dots, Q.
\end{aligned}$$

Come già anticipato, il troncamento iniziale della successione di coefficienti Autoregressivi $\{\pi_i\}$ ad un opportuno lag m non crea particolari problemi nella valutazione di \mathbf{B} dal momento che la invertibilità del processo impone che essi decrescano in valore assoluto al divergere dell'indice i .

Per quanto concerne, poi, la determinazione del valore critico da utilizzare nel test, la distribuzione di una forma quadratica di variabili casuali gaussiane, come definita dalla (4), può essere valutata in maniera esatta sfruttando uno degli algoritmi disponibili (per esempio, le soluzioni di Imhof, 1961; Rice, 1980; Farebrother, 1990) che effettuano, numericamente, l'inversione della corrispondente funzione caratteristica.

Per semplificare, però, l'utilizzo di \hat{d}^2 nelle applicazioni reali si può far ricorso ad una opportuna approssimazione di tale distribuzione (si veda, ad esempio, Mathai e Provost, 1992, per una estesa rassegna). Nel seguito, tra le soluzioni da implementare che risultano semplici ma

sufficientemente accurate, abbiamo ipotizzato che la distribuzione di \hat{d}^2 , sotto H_0 , possa essere approssimata con la distribuzione di una variabile casuale del tipo: $(a\chi_\nu^2 + b)$, in cui χ_ν^2 denota una variabile casuale Chi-quadrato con ν gradi di liberta' (ν non e' necessariamente intero), mentre i parametri (a, b, ν) sono determinati con il metodo dei momenti.

In particolare, posto $t_k = 2^k \text{tr}(\Sigma^k)$ le soluzioni sono:

$$\hat{a} = t_3/t_2; \quad \hat{b} = t_1 - t_2^2/t_3; \quad \hat{\nu} = t_2^3/t_3^2. \quad (7)$$

Per pervenire a tali stime, non e' necessario il calcolo degli autovalori della matrice Σ , ma solo la traccia di potenze della matrice $(B'A'ABV)$ che ha dimensioni $(p + q + P + Q) \times (p + q + P + Q)$, essendo $\text{tr}(\Sigma^k) = \text{tr}[(B'A'ABV)^k]$. Si noti, comunque, che tale calcolo e' richiesto solo per $k = 1, 2, 3$.

D'altra parte, come e' usuale nella teoria della stima, nelle applicazioni reali la matrice Σ non e' nota e va quindi specificata sulla base delle stime $\hat{\beta}_x$ e $\hat{\beta}_y$. A tal fine, si puo' sostituire nella (6) la matrice stimata: $\hat{\Sigma} = 0.5 (\hat{\Sigma}_x + \hat{\Sigma}_y)$, in cui $\hat{\Sigma}_x$ e $\hat{\Sigma}_y$ sono ottenute sostituendo le stime dei parametri dei modelli nella espressione prima individuata per la matrice delle varianze e covarianze.

Infine, si osservi che, per finalita' diverse dal confronto di serie storiche e dalla classificazione, il test puo' essere condotto in forma diversa al fine di verificare se serie osservata e' stata generata da un particolare modello teorico \mathcal{M}_0 . Tale problema risulta di particolare interesse, ad esempio, nell'ambito dello studio delle procedure di depurazione stagionale, per le quali e' nota la loro ottimalita' per particolari strutture generatrici dei dati (Corduas e Piccolo, 1999).

In questo caso, il problema viene posto considerando l'ipotesi $H_0: \pi_x = \pi_0$, (ovvero, $H_0: \beta_x = \beta_0$) e, inoltre, osservando che sotto H_0 , $(\hat{\pi}_x - \pi_0) \stackrel{\Delta}{\sim} \mathcal{N}(\mathbf{0}, \Sigma)$, dove $\Sigma = ABV_0B'A'$.

Dalla trasformazione: $\boldsymbol{\delta} = (\boldsymbol{\Sigma})^{-1/2}(\hat{\boldsymbol{\pi}}_x - \boldsymbol{\pi}_0) \stackrel{a}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$ si avra' che:

$$\widehat{d}^2 = (\hat{\boldsymbol{\pi}}_x - \boldsymbol{\pi}_0)' (\hat{\boldsymbol{\pi}}_x - \boldsymbol{\pi}_0) = \boldsymbol{\delta}' \boldsymbol{\Sigma} \boldsymbol{\delta} = \sum_{j=1}^s \lambda_j \chi_{g_j}^2, \quad (8)$$

in cui, come in precedenza, λ_j sono gli autovalori positivi della matrice $\boldsymbol{\Sigma}$ mentre $\chi_{g_j}^2$ sono variabili casuali Chi-quadrato indipendenti con g_j gradi di liberta'. Anche l'approssimazione con la distribuzione di una variabile casuale del tipo: $(a\chi_{\nu}^2 + b)$, resta valida, tenuto conto che in questo caso: $t_k = tr(\boldsymbol{\Sigma}^k)$.

Nel paragrafo 6 viene presentato lo schema della procedura realizzata in linguaggio GAUSS (Aptech System, 1998) di cui in Appendice riportiamo l'articolazione completa e dettagliata.

5. *L'analisi della matrice delle distanze*

Nella procedura in linguaggio GAUSS, che presenteremo nel prossimo paragrafo, sono state implementate alcune proposte sviluppate da Corduas (1984; 1990) e Piccolo (1984b) per l'analisi della matrice delle distanze.

In particolare, supponiamo di operare su un insieme di k serie storiche per le quali sono disponibili le stime dei modelli ARIMA per esse identificati e stimati. Fra l'altro, dalle successioni di coefficienti Autoregressivi per ciascun modello, e' possibile calcolare alcuni indici:

$$I_r = \sum_i |\pi_i|, \quad r = 1, 2; \quad M_r = \sum_i i^r \pi_i, \quad r = 0, 1, 2$$

i quali costituiscono delle misure del contenuto informativo del processo (Corduas, 1990).

Inoltre, considerando ciascuna serie storica come un 'soggetto' di una analisi multivariata e caratterizzando ciascun soggetto mediante la

successione dei coefficienti Autoregressivi, con una *tecnica di scaling*, si puo' pervenire ad una 'mappa' dei soggetti esaminati. In particolare, Piccolo (1984b) ha proposto l'uso del Multidimensional Scaling metrico (MDS), che consente di ricostruire (sulla base di una matrice di distanze intercorrenti tra k soggetti) la rappresentazione in \mathbb{R}^{k-1} che puo' esserne all'origine. Di seguito, ne sintetizziamo gli aspetti essenziali, che sono stati poi implementati nella procedura discussa nel paragrafo successivo.

Data una matrice di distanze $D = \{d_{ij}\}$ di dimensioni $(k \times k)$, si assume che esista una configurazione di k punti individuati dalla matrice delle coordinate P , rappresentabile al piu' in uno spazio \mathbb{R}^{k-1} e tale che le distanze intercorrenti siano esattamente quelle della matrice di partenza D (Mardia et al., 1979).

Si puo' dimostrare che esiste un insieme di coordinate reali che risponde alle condizioni indicate se e' semidefinita positiva la matrice: $G = \{g_{ij}\}$ i cui elementi sono definiti dalle relazioni:

$$g_{ij} = a_{ij} - \bar{a}_{.j} - \bar{a}_{i.} + \bar{\bar{a}}$$

avendo posto:

$$a_{ij} = -\frac{1}{2}d_{ij}^2; \quad \bar{a}_{.j} = \frac{1}{k} \sum_i a_{ij}; \quad \bar{a}_{i.} = \frac{1}{k} \sum_j a_{ij}; \quad \bar{\bar{a}} = \frac{1}{k^2} \sum_i \sum_j a_{ij}.$$

Poiche', ovviamente, gli autovalori $\lambda_1 > \lambda_2 > \dots > \lambda_k > 0$ di G sono non-negativi, si puo' considerare la decomposizione spettrale:

$$G = \Gamma \Lambda \Gamma'$$

dove $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_k)$ e Γ e' la matrice dei corrispondenti autovettori di G . Posto: $P = \Gamma \Lambda^{1/2}$, possiamo scrivere:

$$G = P P'$$

e le righe della matrice P costituiscono cosi' le coordinate cercate.

E' utile, per comprendere operativamente il risultato a cui si perviene, considerare il seguente schema:

autovalori					
λ_1	λ_2	λ_{k-1}	λ_k	
p_{11}	p_{12}		$p_{1,k-1}$	$p_{1,k}$	coordinate soggetto S_1
p_{21}	p_{22}		$p_{2,k-1}$	$p_{2,k}$	coordinate soggetto S_2
.
p_{k1}	$p_{k,2}$		$p_{k,k-1}$	$p_{k,k}$	coordinate soggetto S_k

$$\bar{p}_{.1} = \bar{p}_{.2} = \dots = \bar{p}_{.k-1} = \bar{p}_{.k} = 0.$$

dove la j -esima colonna e' fornita dall'autovettore di G , corrispondente all'autovalore λ_j , scalato in modo che il prodotto interno dell' j -esimo autovettore e' λ_j ; la i -esima riga fornisce, invece, le coordinate dell' i -esimo soggetto.

A tale riguardo, occorre osservare che:

- i) la rappresentazione MDS e' centrata intorno all'origine degli assi;
- ii) la soluzione MDS e' invariante per rotazione, traslazione e riflessione;
- iii) le prime r colonne della matrice P individuano le coordinate che consentono di rappresentare i soggetti in uno spazio \mathbb{R}^r , con $r < k$, in particolare tale rappresentazione sara' la proiezione in tale sottospazio della configurazione di punti iniziale ottenuta nello spazio \mathbb{R}^{k-1} .

Gli autovalori $\lambda_1, \dots, \lambda_r$ forniscono l'indicazione del contributo di ciascuna dimensione nel riprodurre la configurazione di punti di partenza. In genere, e' opportuno lavorare in uno spazio di dimensioni ridotte $r = 1, 2, 3$, affinche' si possa successivamente produrre una mappa degli

individui esaminati. Per avere una misura della fedeltà e accuratezza della rappresentazione in \mathbb{R}^r , supponendo $\lambda_r > 0$, sono stati proposti i seguenti indici (Mardia et al., 1979):

$$\alpha_1 = \frac{\sum_{j=1}^r \lambda_j}{\sum_{i=1}^k \lambda_j} 100; \quad \alpha_2 = \frac{\sum_{j=1}^r \lambda_j^2}{\sum_{i=1}^k \lambda_j^2} 100; \quad r = 1, 2, \dots, k - 1.$$

Inoltre, mutuando un indicatore noto nel MDS non metrico, si può effettuare un confronto tra le distanze originarie \hat{d}_{ij} con quelle che sono state ricostruite nella rappresentazione in \mathbb{R}^r , che indicheremo come \tilde{d}_{ij} , pervenendo al cosiddetto *indice di stress*:

$$IS = \left(\sum_i \sum_j (\hat{d}_{ij} - \tilde{d}_{ij})^2 / \sum_i \sum_j \hat{d}_{ij}^2 \right)^{1/2}.$$

6. Programmi e procedure in linguaggio GAUSS

In questo paragrafo, descriveremo schematicamente due procedure:

i) la prima rivolta alla costruzione della matrice di distanze D tra i modelli stimati su un insieme di k serie storiche (DISTANZA.PRG), alla valutazione di indici statistici sulle distanze ed alla costruzione della rappresentazione MDS per la loro classificazione;

ii) la seconda finalizzata alla costruzione del test delle ipotesi per il confronto di due serie storiche (TEST2TS.PRG).

Entrambi i programmi presentano una gestione dell'input interattiva cosicché l'utente può agevolmente inserire i dati richiesti. Nel caso di analisi estensive la fase di input è facilmente modificabile dall'utente.

Nella Tabella 1 e 2 sono descritti gli obiettivi del programma principale e di ciascuna procedura, nonché l'input ed output ad esse corrispondenti.

L'Appendice a questo lavoro riporta integralmente i listati dei due programmi, con le corrispondenti procedure, sviluppati in linguaggio GAUSS.

Tabella 1. Elementi essenziali del programma DISTANZA.PRG

Procedure	Descrizione	Input	Output
POLINV	Effettua l'inversione di un polinomio	COEFF, NITER	QUOZ
PRODOTTO	Effettua la moltiplicazione di due polinomi: il primo stagionale ed il secondo non stagionale	SEAPAR,NOSEAPAR	PARAM
DISTA	Calcola la matrice delle distanze	PIWEI	DIST
PINDEX	Calcola gli indici statistici descritti nel paragrafo 5	PIWEI	
MDS	Calcola le coordinate MDS	DIST, CSTA	COORD VDCA
Programma Principale DISTANZA.PRG	Gestisce l'acquisizione dati e le procedure precedenti e i files di risultati	<i>files di dati:</i> DIFF.DAT NOSTAG.DAT STAG.DAT <i>variabili:</i> NMOD ARMA NPAI, MAXORD S, MAXORDS	

Tabella 2. Elementi essenziali del programma TEST2TS.PRG

Procedure	Descrizione	Input	Output
BUILDSEA	Costruisce un polinomio stagionale	VSEA,S	VFIN CVETS
CALCBMAT	Calcola la matrice delle derivate B, i coefficienti AR, la matrice delle varianze e covarianze dei coefficienti	P,Q,PS,QS,S PHI,THETA, PHISES,THETASEA NPIGRE	BMAT PIWEIO MCOV
CALCAMAT	Calcola la matrice A	d,DS,S,NPIGRE	A
CALCTRAC	Calcola le tracce $\text{tr}(2^k \Sigma^k)$, $k = 1, 2, 3$ se il confronto e' tra due modelli stimati	Ax,Bx,Vx,Ay,By,Vy	t1,t2,t3
CALCTRACM0	Calcola le tracce $\text{tr}(\Sigma^k)$, $k = 1, 2, 3$ se il confronto e' con un modello teorico	Ax,Bx,Vx	t1,t2,t3
Programma Principale TEST2TS.PRG	Gestisce l'acquisizione dati e le procedure precedenti	<i>variabili:</i> TIPO, s px,dx,qx,Psx,Dsx,Qsx,nx py,dy,qy,Psy,Dsy,Qsy,ny	percentile asintotico

7. Considerazioni finali

Lo sviluppo di una procedura operativa che consente di implementare le proposte metodologiche in relazione alla metrica Autoregressiva costituisce un elemento importante per una diffusione operativa di tale approccio.

In questo lavoro, abbiamo discusso i passi essenziali per costruire un programma di calcolo in linguaggio GAUSS sfruttando le particolari capacita' che tale ambiente di programmazione mette a

disposizione in termini di calcolo matriciale. Infine, l'Appendice presenta il listato completo dei programmi e delle procedure discusse.

Ringraziamenti: Lavoro realizzato con parziale finanziamento da fondi di Ateneo e MURST erogati per il Progetto di Ricerca di interesse nazionale: "Modelli statistici per l'analisi delle serie temporali".

Riferimenti bibliografici

Aptech System (1998) *GAUSS Reference Manual*, Mapley Valley (WA).

Box G.E.P., Jenkins G.M. (1976) *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.

Brockwell P.J., Davis R.A. (1987) *Time Series Theory and Methods*, Springer-Verlag, New York.

Corduas M. (1984) Distanza tra modelli: problemi metodologici ed indici statistici, *Statistica*, XLIV, 513-524.

Corduas M. (1990) Indicatori strutturali per processi stocastici lineari, *Statistica*, XLIX, 2, 1-16

Corduas M. (1992) On the distributional properties of a distance criterion for time series models, *Atti del Convegno 'DISTANCIA 92'*, Rennes, 325-328.

Corduas M. (1993) An approximation to the distribution of the Euclidean distance between Moving Average models, *Bullettin of the International Statistical Institute (49th Session)*, Firenze, vol.I, 283-284.

Corduas M. (1996) Uno studio sulla distribuzione asintotica della metrica Autoregressiva, *Statistica*, LVI, 321-332.

Corduas M., Piccolo D. (1996) Time series clustering of the Italian Consumer Price Indices: a model approach, *Quaderni di Ricerca ISTAT*, Roma.

- Corduas M., Piccolo D. (1999) On the use of AR metric for seasonal adjustment , *Atti del Convegno "CLADAG-99"*, Roma
- Farebrother R.W. (1990) The distribution of a quadratic form in Normal variables, *Applied Statistics*, 294-309.
- Imhof P.J. (1961) Computing the distribution of quadratic forms in Normal variables, *Biometrika*, 48, 419-426.
- Maharaj E.A. (1996) A significance test for classifying ARMA models, *Journal of Statistical Computation and Simulation*, 54, 305-331
- Mardia K.V., Kent J.T., Bibby J.M. (1979) *Multivariate analysis*, Academic Press, New York.
- Mathai A.M, Provost S.B. (1992) *Quadratic forms in random variables*, Marcell Dekker Inc., New York.
- Piccolo D. (1984a) Una topologia per la classe ARIMA, *Statistica*, XLIV, 47-59.
- Piccolo D. (1984b) Una rappresentazione multidimensionale per modelli statistici dinamici, *Atti della XXXII Riunione Scientifica SIS*, Sorrento, vol.I, 149-160
- Piccolo D. (1989) On the measure of dissimilarity between ARIMA models, *Proceedings of the A.S.A. Meetings, Business and Economic Statistics Sect.*, Washington D.C., 231-236.
- Piccolo D. (1990) A distance measure for classifying ARIMA models, *Journal of Time Series Analysis*, 11, 153-164.
- Rice S.O. (1980) Distribution of quadratic forms in Normal variables: evaluation by numerical integration, *SIAM Journal of Scientific and Statistical Computing*, 1, 438-448.

Appendice

Nota: La circolazione e diffusione dei seguenti programmi e' libera e gratuita purché si citi correttamente ed integralmente la fonte. Resta dell'utente ogni responsabilita' per i danni derivanti dall'utilizzo dei programmi qui descritti.

©Corduas, M. (2000)

PROGRAMMA: DISTANZA.PRG

@

PROC PRODOTTO: Costruisce il polinomio stagionale espandendo ed inserendo zero ai lags intermedi. Effettua il prodotto tra due polinomi: il primo stagionale ed il secondo non stagionale.

INPUT: due matrici nelle cui righe sono riportati i coefficienti dei polinomi da moltiplicare compreso 1 e gli zeri e con segni ed ordinamento come in $(1+a*x + b*x^2....)$

OUTPUT: matrice dove per riga sono riportati i polinomi prodotto definiti con segni + ed in ordine discendente

@

```
Proc PRODOTTO (seapar,noseapar);
local j,s,ncols,nns,rs,zeta,mats,colonne,param,nrighe,a,b,c;
  s=varget("s");
  ncols= cols(seapar);
  nns= (ncols-1)*s ;
  rs= rows(seapar) ;
  zeta= zeros(rs,s-1) ;
if ncols ==2;
  mats=seapar[.,1]~zeta~ seapar[.,2] ;
else;
  mats=seapar[.,1]~zeta~seapar[.,2]~zeta~seapar[.,3] ;
endif;
colonne=cols(noseapar)+cols(mats)-1;
param=zeros(1,colonne);
nrighe=rows(seapar);
j=1;
do while j<=nrighe;
  a=rev(mats[j,.]') ;
  b=rev(noseapar[j,.]');
  c=conv(a,b,0,0);
  param=param|c';
j=j+1;
```

```

j=j+1;
endo;
  param=trimr(param,1,0);
  param=rev(param)';
  retp(param);
endp;

```

@

PROC POLINV: Effettua l'inversione di un polinomio
INPUT: vettore riga dei coefficienti dal grado 0 in modo ascendente
OUTPUT: vettore del polinomio inverso, i coefficienti includono i segni
ESEMPIO: (1-0.5B)
INPUT: 1⁻ -0.5
OUTPUT: 1 .5 .25 .625 ETC..

@

```

Proc POLINV(coeff,niter);
local j,r,quoz ;
coeff=coeff';
r=rows(coeff);
quoz=zeros(niter,1);
coeff=coeff|zeros(niter-r,1);
quoz[1,1]=coeff[1,1];
quoz[2,1]=coeff[2,1];
j=3;
do while j<=niter;
  quoz[j,1]=coeff[j,1] - quoz[2:j-1,.*]*rev(coeff[2:j-1,.*]);
j=j+1;
endo;
  quoz[2:rows(quoz)]=-quoz[2:rows(quoz)];
  retp(quoz);
endp;

```

@

PROC PINDEX: Calcola gli Indici sui Pi-Weights

@

```

Proc PINDEX(piwei);
local vv,index,sp,sp2,spabs,sjp,sj2p,m2,mat,r,vetj;
r=rows(piwei);
mat=piwei[2:r,.*];
sp=sumc(mat);

```

```

sp2=sumc(mat^2);
spabs=sumc(abs(mat));
vetj=seqa(1,1,r-1);
sjp=sumc(vetj.*mat);
sj2p=sumc((vetj^2).*mat);
m2=sp2./(1+sp2);
vv=seqa(1,1,rows(sp));
index=vv~sp~sp2~spabs~sjp~sj2p~m2 ;
format /m2 /rd 10,5;
print
" modello somma pi som pi^2 som abs som j*pi som j^2*pi m2";
print index;
endp;

```

@

PROC DISTA: CALCOLA LA MATRICE DELLE DISTANZE EUCLIDEE

@

```

Proc DISTA(piwei);
local dist,i,j,nmod,indvar;
  nmod=cols(piwei);
  dist=zeros(nmod,nmod);
i=1;
  do while i<=nmod-1;
    j=i+1;
    do while j<=nmod;
      dist[i,j]=sumc((piwei[.,i]-piwei[.,j])^2);
      j=j+1;
    endo;
    i=i+1;
  endo;
dist=dist+dist';
indvar=sumc(dist)/(nmod-1);
dist=sqrt(dist);
format /m2 /rd 9,0;
print "matrice distanze euclidee";
print "-----";
print "cols";
print seqa(1,1,nmod)';
print "-----" "\|";
format /m3 /rd 9,4;
print dist;
print "-----" ;
?;?;
print "\| "indice di variabilita' ";?;print indvar';

```

```
retp(dist);
endp;
```

```
@
PROC MDS: EFFETTUA L'ANALISI MDS SU UNA MATRICE DI DISTANZE
INPUT: MATRICE DELLE DISTANZE, CSTA (CONTROLLA LE STAMPE)
OUTPUT: MATRICE DELLE COORDINATE, VETTORE DELLE DISTANZE
ORIGINALI, VETTORE DELLE NUOVE DISTANZE-MDS
@
```

```
Proc(2)= MDS(distan,csta);
local g,r,h,aval, avetr,matcom, avalt,indabs,ind2,coord, distco,nc, stress,vdico;
@-----costruisce matrice G-----@
g=-0.5*distan^2;
r=rows(distan);
h=eye(r)-(ones(r,r)/r);
g=h*g*h;
@---calcola autovalori e autovettori-----@
clearg avetr,aval;
{aval,avetr}=eighv(b);
avetr=avetr./sqrt((sumc(avetr^2)));
@-----ordina gli autovalori e gli autovettori-----@
matcom=(aval[.,1]|avetr)';
matcom=rev(sortc(matcom,1));
@-----calcola indici e le coordinate-----@
avalt=matcom[.,1];
indabs=100*cumsumc(abs(avalt))./sumc(abs(avalt));
ind2=100*cumsumc(avalt^2)./sumc(avalt^2);
matcom=selif(matcom,matcom[.,1].gt 0.00001);
coord=(matcom[.,2:r+1])*diagrv(eye(rows(matcom)),sqrt(matcom[.,1]));
distco=zeros(r,r);
print "\f";
print "-----";
print "a n a l i s i m d s " "\f";
print "-----";
print " autovalori        alfa1        alfa2 ";
print "-----";
format /m1 /ro 13,3;
print "\f" avalt~indabs~ind2;
print "-----";
?;?;?;
format /m2 /rd 13,4;
if (csta $== "n"); goto sub2; endif;
print "coordinate" "\f";
if cols(coord)>=3; nc=3; else; nc=2; endif;
coord=coord[.,1:nc];
```

```

@considera i primi nc assi@
print seqa(1,1,rows(coord))~coord[:,1:nc];?;?;?;
print avalt[1:nc,1]' " autovalori" ;
print indabs[1:nc,1]' " alfa1" ;
print ind2[1:nc,1]' " alfa2";
@-----calcola distanze nel nuovo spazio, stress-----@
sub2:
print "\f" "analisi sulle coordinate mds";?;?;
distco=dista(coord[:,1:2]);
stress=sqrt((sumc(sumc((distan-distco).^2)))/(sumc(sumc(distan^2))));
?;?;
print "indice di stress tra la configurazione originale e quella ricostruita";
print "sqr{ somma[d(i,j)-d^(i,j)**2]/(somma d(i,j)**2)}=" stress; ?;?;?;
vdico=vec(distco);
print "\f";
@----retp matrice coordinate, vec(matrice nuove distanze)-----@
retp(coord,vdico);
endp;

```

@-----CARICA MATRICE DIFFERENZE-----@

```

LET MATDIF[5,14]=  1 0 0 0 0 0 0 0 0 0 0 0 0 0
                  1 -1 0 0 0 0 0 0 0 0 0 0 0 0
                  1 0 0 0 0 0 0 0 0 0 0 -1 0
                  1 -1 0 0 0 0 0 0 0 0 0 -1 1
                  1 2 1 0 0 0 0 0 0 0 0 0 0 0 ;

```

@

:PROGRAMMA PRINCIPALE: CALCOLA I COEFFICIENTI PI-GRECO DI NMOD
MODELLI NELLA LORO STRUTTURA ARMA (oppure ARIMA)

OCCORRE FORNIRE TRE FILES:

DIFF.DAT: contiene per ciascun modello i valori:

- 1 se la serie non e' differenziata
- 2 per DELTA(X[t])
- 3 per DELTA2(X[t])
- 4 per DELTA12(X[t])
- 5 per DELTA DELTA12(X[t])

NOSTAG.DAT: contiene 2 righe per ciascun modello, la prima per l'operatore AR e
la seconda per l'operatore MA non stagionale

STAG.DAT: contiene 2 righe per ciascun modello, la prima per l'operatore AR
e la seconda per l'operatore MA stagionale

-i valori dei coefficienti sono indicati sempre comprensivi di segno, l'operatore e'
percio' definito come $1+a_1 B+a_2 B^2\dots$

- in caso un coefficiente non sia presente va posto nella riga uno zero

-in caso di assenza di un operatore il file contiene una riga di zeri

ESEMPIO: $(1-0.3B^{12})\Delta(X[t])=(1+0.2B)(1-0.6B^{24})a[t]$

DIFF.DAT--->contiene: 2

NOSTAG.DAT---->contiene due righe l'ordine max e' 2

0 -0.3
0.2 0

STAG.DAT ---->contiene:

0 0
0 -0.6

NOTA: L'OUTPUT VERRA' SCARICATO IN UN FILE LA CUI DENOMINAZIONE
E' FORNITA DALL'UTENTE

@

"numero modelli";

nmod=con(1,1);

@ numero di modelli da analizzare

@

"analisi arma=1; arima=0";

arma=con(1,1);

@ =0 effettua l'analisi sui modelli arima,

=1 effettua l'analisi sulla sola parte arma@

"ordine massimo operatori non stagionali";

maxord=con(1,1);

@ ordine non stagionale massimo di tutti i modelli @

"stagionalita' ";

s=con(1,1);

"ordine massimo operatori stagionali";

maxords=con(1,1);

@ ordine stagionale massimo tra tutti i modelli@

"numero di coefficienti pi-greco";

npai=con(1,1);

@ numero di pigreco da calcolare @

"subdirectory ove si opera es. c:\gauss\distanze";?;

subdire=upper(cons);

"nome del file su cui salvare i risultati (n.b.: se il file gia' esiste viene ricoperto);?;

usc=upper(cons);

usc=subdire \$+ "\" \$+ usc;

output file=^usc reset;

@apre file per la stampa @

"effettua mds? (s/n)";

conmds=upper(cons);

@ controlla esecuzione mds@

if (conmds \$== "s");

"stampa le coordinate (s/n) ";

csta=upper(cons);

endif;

```

"salva i coefficienti pi-greco? (s/n);
    consalva=upper(cons);
if (consalva $== "s");
    "inserire nome del file: ";
    uscpi=upper(cons);
    uscpi=subdire $+ "\" $+ uscpi;
endif;

format /m2 /rd 8,3;
print "controllo input";?;
print "nmod= " nmod "\" arma= " arma;
print "maxord= " maxord "\" "maxords=" maxords;?;

if arma==0;
    print "\" "a n a l i s i a r i m a" "\";
    nome1=subdire $+ "\"diff.dat";
    load opdiff[nmod,1]^nome1;
    dim=maxord+(maxords*s)+13;
    dimpi=dim+npai;
else;
    print "\" "a n a l i s i a r m a" "\";
    dimpi=maxord+(s*maxords)+npai;
endif;
nome1=subdire $+ "\"nostag.dat";
load nostag[2*nmod,maxord]^nome1;
nostag=ones(2*nmod,1)~nostag;

print "controllo parametri non stagionali" "\" nostag;?;
print "\f";

@--se non vi sono oper.stagionali salta al calcolo pigreco-----@

clear matpro;
if s==0;
    matpro=nostag;
else;
    nome1=subdire $+ "\"stag.dat";
    load stag[2*nmod,maxords]^nome1;
    stag=ones(2*nmod,1)~stag;

print "controllo parametri stagionali" "\" stag;?;?; print "\f";

    matpro=prodotto(stag,nostag);
endif;

rigar=seqa(1,2,nmod);
rigma=rigar+1;
compar=matpro[rigar,.];

```

```

compma=matpro[rigma,.];
quoz=zeros(nmod,npai);

@                               inverta polinomio theta  calcola delta*phi se analisi arima
                               calcola coefficienti pigreco                               @

piwei=zeros(1,dimpi);
clear com;
j=1;
do while j<=nmod;
j;
    quoz[j,.]=polinv(compma[j,.],npai);
    if arma==1;
        comparg=compar[j,.];
        comparg=rev(comparg);
    else;
        ind=opdiff[j,1];
        revdif=rev(matdif[ind,.]);
        revcom=rev(compar[j,.]);
        comparg=conv(revdif,revcom,0,0);
    endif;

                               @          calcola coefficienti pigreco          @
        com=conv(comparg,rev(quoz[j,.]),0,0);
        mm=minc(dimpi|rows(com));
        piwei=piwei[.,1:mm][com[1:mm]'];

j=j+1;
endo;

        piwei=trimr(piwei,1,0);
        piwei=(rev(piwei'));
        piwei=-piwei[1:npai,.];

if (consalva $== "s");
    output file=^uscpi reset;
    print " @ pi-weights @ ";
    print seqa(0,1,npai)~piwei;
endif;
call pindex(piwei);
print "\f";
distan=dista(piwei);
if (conmds $== "s");
    {coo,vdi}=mds(distan,csta);
endif;
format /m1 /rd 9,6;
output off;
end;

```

PROGRAMMA: TEST2TS.PRG

```
/* _____
TEST delle IPOTESI:
SE TIPO=1;
    H0: pigrecox=pigrecoy utilizza  $d^2 = \text{Somma}(\text{pix}^2 - \text{piy}^2)$ 

INPUT: per ciascun modello
ordini del modello: px,dx,qx,PSx,DSx,QSx,Sx, s; valore dei coefficienti stimati,
matrice delle varianze e covarianze di tali stime Vx, num.pigreco
OUTPUT:  $d^2$  tra i due modelli, percentile asintotico al 5%, utilizzando
l'approssimazione  $[a \chi^2(\text{nu}) + b]$ 
NB: I segni con cui sono definiti gli operatori sono quelli di Box e Jenkins
Le matrici di varianze e covarianze dei coefficienti stimati vanno ricavate durante la
fase di stima

SE TIPO=0;
confronta un modello stimato Mx con uno teorico M0
H0: pigrecox=pigreco0 utilizza  $d^2 = \text{Somma}(\text{pix}^2 - \text{pi0}^2)$ 

INPUT:
per il modello teorico
ordini del modello: p,d,q,PS,DS,QS,S, s; valore dei parametri, matrice delle
varianze e covarianze Vx
per il modello stimato
ordini del modello: px,dx,qx,PSx,DSx,QSx,Sx, s; valore dei coefficienti stimati,
num.pigreco

OUTPUT:  $d^2$  tra i due modelli, percentile asintotico al 5%, all'1% utilizzando
l'approssimazione  $[a \chi^2(\text{nu}) + b]$ 
/* _____

proc(2)=buildsea(vsea,s);
local nums,cvets,vfin,i;
nums=rows(vsea);
vfin=1;
cvets=0;
    i=1;
    do while i<=nums;
        vfin=vfin|zeros(s-1,1)|-vsea[i];
        cvets=cvets|zeros(s-1,1)|-1;
        i=i+1;
    endo;
retp(vfin,cvets);
```

endp;

```
/* _____  
CALCBMAT: --->CALCOLA LA MATRICE DELLE DERIVATE B  
INPUT: il modello  
ordini del modello: px,dx,qx,PSx,DSx,QSx,Sx, s; valore dei coefficienti stimati,  
num.pigreco  
OUTPUT: i coefficienti pi-greco; la matrice delle derivate B, la matrice delle varianze  
e covarianze dei coefficienti  
_____ */
```

```
proc(3)=CALCBMAT(p,q,ps,qs,s,phi,theta,phisea,thetasea,npigre,ndati);  
local vphi, vphisea, cvphis, f, vthe, lenne, ilenne, vthetasea, cvthes, ls, ils, ll,  
mf, piweio, piweino, bmat, h, bvet,deltan, deltaseaf, deltaseat, aa, indici,  
invphi,invthe,invthes,invphis,matde,mphi,mthe,mphis,mthes,mcov;  
  
matde=zeros(npigre,1); @inizializza matrice var covar@  
if p > 0;  
    vphi=-1|phi|zeros(npigre-p-1,1);  
    invphi=polinv(1|-phi,npigre);  
    mphi=lowmat(toeplitz(invphi));  
    mphi=mphi[.,2:p+1];  
    matde=matde~mphi;  
else;  
    vphi=-1|zeros(npigre-1,1);  
endif;  
if q>0;  
    invthe=polinv(1|-theta,npigre);  
    vthe=1|-theta|zeros(npigre-q-1,1);  
    lenne=lowmat(toeplitz(vthe));  
    ilenne=inv(lenne);  
    mthe=lowmat(toeplitz(invthe));  
    mthe=-mthe[.,2:q+1];  
    matde=matde~mthe;  
else;  
    lenne=eye(npigre);  
    ilenne=eye(npigre);  
endif;  
if ps>0;  
    {vphisea, cvphis}=buildsea(phisea,s);  
    invphis=polinv(vphisea,npigre);  
    vphisea=vphisea|zeros(npigre-ps*s-1,1);  
    cvphis=cvphis|zeros(npigre-ps*s-1,1);  
    f=lowmat(toeplitz(vphisea));  
    mphis=lowmat(toeplitz(invphis));
```

```

        mphis=mphis[.,s:ps+s-1];
        matde=matde~mphis;
else;
    f=eye(npigre);
endif;
if qs>0;
    {vthesea,cvthes}=buildsea(thetasea,s);
    invthes=polinv(vthesea,npigre);
    vthesea=vthesea|zeros(npigre-qs*s-1,1);
    cvthes=cvthes|zeros(npigre-qs*s-1,1);
    ls=lowmat(toeplitz(vthesea));
    ils=inv(ls);
    mthes=lowmat(toeplitz(invthes));
    mthes=-mthes[.,s:qs+s-1];
    matde=matde~mthes;
else;
    ls=eye(npigre);
    ils=eye(npigre);
endif;

matde=matde[.,2:p+q+ps+qs+1];           @calcola matrice var e cov @
matde=matde'matde;
mcov=inv(matde)/ndati;
ll=ils*ilenne;
mf=ll*f;
piweio=mf*vphi;
piweino=ilenne*vphi;
bmat=zeros(npigre,1);
    if p==0;
bmat=bmat~zeros(npigre,1);
else;
    h=1;
    do while h<=p;
        if vphi[h+1] .eq 0;
            bmat=bmat~zeros(npigre,1);
        else;
            bvet=zeros(npigre,1);
            bvet[h+1]=1;
            bmat=bmat~(mf*bvet);
        endif;
        h=h+1;
    endo;
endif;
if q==0;
bmat=bmat~zeros(npigre,1);
else;
    h=1;
    do while h<=q;

```

```

        if vthe[h+1] .eq 0;
            bmat=bmat~zeros(npigre,1);
        else;
            deltan=-eye(npigre);
            deltan=shiftr(deltan,h,0)';
            bmat=bmat~(-ilenne*deltan*piweio);
        endif;
    h=h+1;
endo;
endif;
if ps==0;
bmat=bmat~zeros(npigre,1);
else;
    h=1;
    do while h<=ps;
        if vphisea[h+1] .eq 0;
            bmat=bmat~zeros(npigre,1);
        else;
            deltaseaf=-eye(npigre);
            deltaseaf=shiftr(deltaseaf,h,0)';
            bmat=bmat~(l1*deltaseaf*vphi);
        endif;
    h=h+1;
    endo;
endif;
if qs==0;
bmat=bmat~zeros(npigre,1);
else;
    h=1;
    do while h<=qs;
        if vthesea[h+1] .eq 0;
            bmat=bmat~zeros(npigre,1);
        else;

            deltaseat=-eye(npigre);
            deltaseat=shiftr(deltaseat,h,0)';
            bmat=bmat~(-ilenne*ils*deltaseat*ils*f*vphi);
        endif;
    h=h+1;
    endo;
endif;
aa=seqa(1,1,cols(bmat))~sumc(abs(bmat));
indici=selif(aa,aa[.,2] .ne 0);
indici=indici[.,1];
bmat=bmat[.,indici];
retp(bmat,mcov,piweio);
endp;

```

```

/* _____
CALCAMAT: --->CALCOLA LA MATRICE A per la trasformazione dei coefficienti
ARMA nei coefficienti pigreco dell'ARIMA
INPUT: d=0 oppure 1 DS=0 oppure 1, s=stagionalita', num.pigreco
OUTPUT: la matrice A
_____ */

```

```

proc(1)=CALCAMAT(d,DS,s,npigre);
local A,Ad,ADS,AADS;
if ((d .eq 0) .and (DS .eq 0));
    A=eye(npigre);
elseif ((d .eq 1) .and (DS .eq 0));
    A=-eye(npigre);
    A=shiftr(A,d,0)'+eye(npigre);
elseif ((d .eq 0) .and (DS .eq 1));
    A=-eye(npigre);
    A=shiftr(A,DS*s,0)'+eye(npigre);
elseif ((d .eq 1) .and (DS .eq 1));
    A=-eye(npigre);
    Ad=shiftr(A,d,0)';
    ADS=shiftr(A,DS*s,0)';
    AADS=shiftr(-A,DS*s+1,0)';
    A=eye(npigre)+Ad+ADS+AADS;
endif;
retp(A);
endp;

```

```

/* _____
CALCTRAC: --->CALCOLA LE TRACCE t1, t2,t3
(vedi articolo STATISTICA, 1996)
INPUT: matrici A,B,V ottenute per i due modelli
OUTPUT: tracce t1, t2,t3
_____ */

```

```

proc(3)=CALCTRAC(Ax,Bx,Vx,Ay,By,Vy);
local t1,t2,t3,omegax,omegay,omegac,oo;
omegax=Ax*Bx*Vx*Bx'*Ax';
omegay=Ay*By*Vy*By'*Ay';
omegac=(omegax+omegay);
t1=sumc(diag(omegac));

```

```

oo=omegac*omegac;
t2=sumc(diag(oo));
oo=oo*omegac;
t3=sumc(diag(oo));
retp(t1,t2,t3);
endp;

```

```

/*
-----
CALCTRACM0: --->CALCOLA LE TRACCE t1, t2,t3 (vedi Corduas, 2000)
INPUT:  matrici A,B,V ottenute per il modello M0
OUTPUT: tracce t1, t2,t3
-----
*/

```

```

proc(3)=CALCTRACM0(A,B,V);
local t1,t2,t3,omega,oo;
omega=A*B*V*B*A';
t1=sumc(diag(omega));
oo=omega*omega;
t2=sumc(diag(oo));
oo=oo*omega;
t3=sumc(diag(oo));
retp(t1,t2,t3);
endp;

```

```

/*
-----
Programma principale:
chiede a console l'input
se TIPO=0 considera il caso di distanza da un modello teorico M0 (che e' definito
come modello X)
se TIPO=1 considera il caso di distanza tra due modelli stimati
-----
*/

```

```

"TIPO DI ANALISI: =1 tra due modelli stimati; =0 tra un modello teorico ed uno
stimato";
TIPO=con(1,1);

```

```

"numero pi-greco";
npigre=con(1,1);
if TIPO==1;
    "MODELLO X";
else;
    "MODELLO (teorico) M0";
endif;

```

```

"p";    px=con(1,1);
"d";    dx=con(1,1);

```

```

"q";    qx=con(1,1);
"PS";   psx=con(1,1);
"DS";   dsx=con(1,1);
"QS";   qsx=con(1,1);
"S";    s=con(1,1);
"N.obs."; ndatix=con(1,1);

if px>0;
    " vettore colonna dei coefficienti ";
    phix=con(px,1);
else;
    phix=0;
endif;
if qx>0;
    " vettore colonna dei coefficienti MA ";
    thetax=con(qx,1);
else;
    thetax=0;
endif;
if psx>0;
    " vettore colonna dei coefficienti ARStagionali";
    phiseax=con(psx,1);
else;
    phiseax=0;
endif;
if qsx>0;
    " vettore colonna dei coefficienti MASTagionali ";
    thetaseax=con(qsx,1);
else;
    thetaseax=0;
endif;

"MODELLO Y";
"p";    py=con(1,1);
"d";    dy=con(1,1);
"q";    qy=con(1,1);
"PS";   psy=con(1,1);
"DS";   dsy=con(1,1);
"QS";   qsy=con(1,1);
"N.obs."; ndatiy=con(1,1);
if py>0;
    " vettore colonna dei coefficienti ";
    phiy=con(py,1);
else;
    phiy=0;
endif;
if qy>0;
    " vettore colonna dei coefficienti ";

```

```

        thetay=con(qy,1);
else;
    thetay=0;
endif;
if psy>0;
    " vettore colonna dei coefficienti ARStagionali ";
    phiseay=con(psy,1);
else;
    phiseay=0;
endif;
if qsy>0;
    " vettore colonna dei coefficienti MASTagionali ";
    thetaseay=con(qsy,1);
else;
    thetaseay=0;
endif;

{By,Vy,piweiy}=CALCBMAT(py,qy,psy,qsy,s,phiy,thetay,phiseay,thetaseay,npigre,ndatiy);
    Ay=CALCAMAT(dy,dsy,s,npigre);
    piweiy=Ay*piweiy;

{Bx,Vx,piweix}=CALCBMAT(px,qx,psx,qsx,s,phix,thetax,phiseax,thetaseax,npigre,ndatix);
    Ax=CALCAMAT(dx,dsx,s,npigre);
    piweix=Ax*piweix;

    d2=(piweix-piweiy); d2=d2/d2;
    "d^2= ";; d2;
IF TIPO==1;
    {to1,to2,to3}=CALCTRAC(Ax,Bx,Vx,Ay,By,Vy);
else;
    {to1,to2,to3}=CALCTRACM0(Ax,Bx,Vx);
ENDIF;

ahat=to3/to2;
bhat=to1-(to2*to2/to3);
nuhat=(to2^3)/(to3^2);
pasintot=cdfchii(0.95,nuhat)*ahat+bhat;
"ahat= " ahat;;
"bhat= " bhat;;
"nuhat= " nut;
"percentile al 95% =" ;; pasintot;
end;

```